

# ATR Copy Center#

The ATR Copy Center is available as free software under the [GPLv3 License](#)

[ATR image](#)

## SPL source code:#

```
# ATR Copy Center
# 2008-2014 Carsten Strotmann
# 2014/03/15

const bl $20 ( blank )

const rd $40
const wr $80

const ddevic $300
const dunit $301
const dcmnd $302
const dstats $303
const dbuf $304
const dtimlo $306
const dbyt $308
const daux $30A

const memtop $2E5 ( last free memory location )
const memlo $2E7 ( first free memory location )
const appmhi $0E

var percom 12
var drvstat 17
var buf $100
var dpath $100
var secbuf 2
var bufbase 2
var bufsize 2
var bufend 2
var secbuforg 2
var hsbases 2
var hss2u 2
var hsdriive 2
var siov 2
var siolen 2
var tracks 2
var secpertrack 2
var errorcode 1

var devid 1
var devnm 1
var bytsec 2
var sectors 2
var sector 2
var autoinc 1
var side 1
var i 2
var fcount 2
var cdigit 2
```

```
var sidetgl 2
var cnt 2
var debugflg 1

str stfilen "\FILE0000.ATR"
str sterror " Error"
str stwhile "while"
str streading "reading"
str stwriting "writing"
str stsector "Sector"
str stsize "size"
str stillegal "illegal"
str stcreating "creating"
str stmounting "mounting"
str stremoving "removing"
str statr "ATR"
str stsingle "single"
str stmedium "medium"
str stdouble "double"
str stauto "Auto"
str stincrement "increment"
str stpress "press"
str stkey "key"
str stdrive "Drive"
str ststatus "status"
str sttimeout "Timeout"
str stwd2793 "WD2798"
str stsrc "Src"
str stdest "(D3:) Dst"
str stdensity "Density"
str stdigit "Digit"
str ston "On "
str stoff "Off"
str stmake "make"
str stmount "mount"
str stcopy "Copy"
str stread "Read "
str stwrite "Write"
str stzeros0 "00000"
str stzeros1 "00000"
str stzeros2 "00000"
str stcompleted "completed"
str stsuccessfully "successfully"
str stunsuccessful "unsuccessfully"
str stcenter "Center"
str stdots "..."
str sttracks "Tracks"
str ststep "Step"
str strate "Rate"
str stsectrack "Sec/Track"
str stsides "Sides"
str stbytessec "Bytes/Sec"
str stencoding "Encoding"
str stside "Side"
str sthelpmsg "[H] for help"
str stnextcpy "[space] to start next copy"
str sthighspeed "high speed"
str stnormalspeed "normal speed"
```

```

: debug
  dup 20 20 at "debug:" disp dup .$ " " disp " " disp .
;

: debugk
  dup 20 20 at "debug:" disp dup .$ " " disp " " disp . key drop
;

: dispSPACE
  disp space
;

: init ( -- )
  0 debugflg c!
  $44 devid c!
  1 devnm c!
  $80 bytsec !
  18 secpertrack
  720 sectors !
  1 autoinc c!
  0 side c!
  0 fcount !
  4 cdigit !
  end@ secbuf !
  $E459 hss2u !
  $E459 hsdrive !
  $E459 siov !
  end@ appmhi !
;

: lcls ( y -- )
  crsoff 0 at 156 emit ;

: insline ( y -- )
  0 at 157 emit ;

: clearline ( y -- )
  39 i ! {
    dup i at 31 emit
    i --
    i @ 0 = if break then
  }
  drop
;

: rcls ( pos lines -- )
  i ! {
    dup lcls
    i --
    i @ 0 = if break then
  }
  drop
;

: line ( y -- )
  40 * $58 @ + 40 $52 fill ;

: colon ( -- )
  $3A emit ;

```

```

data bittable
    $01 $02 $04 $08
    $10 $20 $40 $80
end

: bits ( n -- bm )
    bittable + c@ ;

code scopy
[    ldy #5 ]
[    jmp sc1 ]
end
code sc1
[    lda stzeros1,y ]
[    clc    ]
[    adc #1 ]
[    sta stzeros1,y ]
[    cmp #$3A ]
[    bne sc2 ]
[    lda #$30 ]
[    sta stzeros1,y ]
[    dey ]
[    bne sc1 ]
[    jmp sc2 ]
end
code sc2
[    rts    ]
end

: sio ( -- rc )
    siov @ call
    dstats c@
;

: secbuffetch ( -- )
    secbuf @ ;

: clearbuf
    buf $FF $FF fill
    0 buf c!
    0 buf $7F + c!
;

: buf1 ( -- addr )
    buf 1+ ;

: preparesio
    $31 ddevic c!
    devnm c@ dunit c!
    7 dtimlo c!
    0 daux !
;

: writecmd
    wr dstats c!
    sio
;

```

```

: readcmd
  rd dstats c!
  sio
;

: getstatus ( -- rc )
  preparesio
  $53 dcmnd c!
  drvstat dbuf !
  4 dbyt !
  readcmd
;

: getsiolen ( n -- l )
  preparesio
  $68 dcmnd c!
  dunit c!
  secbuf @ dbuf !
  2 dbyt !
  readcmd drop
  secbuf @ @
  dup siolen !
;

: gethssio ( addr n -- )
  dup getsiolen dbyt !
  preparesio
  dunit c!
  $69 dcmnd c!
  dup dbuf ! daux !
  $40 dtimlo c!
  readcmd
;

: gets2usio ( -- )
  appmhi @ 3 gethssio
  1 = if
    appmhi @ hss2u !
    siolen @ appmhi +!
  then
;

: getdrvsio ( -- )
  hsbases @ appmhi !
  appmhi @ devnm @ gethssio
  1 = if
    appmhi @ hsdribe !
    siolen @ appmhi +!
  then
  gets2usio
;

: cprint ( addr -- )
  c@ u. ;

: cprintplus ( addr n -- )
  + cprint ;

: printstatus

```

```

stdrive disp
ststatus disp colon drvstat cprint
" " disp stwd2793 disp colon drvstat 3 cprintplus cr
sttimeout disp colon drvstat 2 cprintplus
;

: mdquery ( tests for medium density )
getstatus drop drvstat c@ $80 and
dup 0 > if
    1040 sectors !
    $80 bytsec !
    26 secpertrack !
then ;

: clearpercom
percom 12 0 fill ;

: getsectors ( -- sec )
percom c@ percom 2 + @ >< *
;

: getpercom ( -- rc )
clearpercom preparesio
$4E dcmnd c!
percom dbuf !
12 dbyt !
readcmd getsectors sectors !
;

: printpercom ( -- )
    sttracks disp colon                percom cprint
cr ststep disp strate disp colon      percom 1 cprintplus
cr stsectrack disp colon               percom 2+ @ >< u.
cr stsides disp colon                  percom 4 cprintplus
cr stencoding disp colon                percom 5 cprintplus
cr stbytessec disp colon                percom 6 + @ >< u.
cr stdrive disp ston disp colon         percom 8 cprintplus
cr stsector disp colon                  getsectors u.
;

# sio2usb code

: temp2dpath
stfilen dpath strcpy
;

: dpath2buf
dpath buf strcpy
;

: s2ucmd
$71 ddevic c! 0 dunit c! $80 dbyt !
;

: s2ucmdp
ddevic c@ $71 =
;

: gets2ustatus

```

```

prepresio s2ucmd
17 dbyt !
drvstat dbuf !
0 dcmd c! # get U status info
readcmd
;

: makeatr
clearbuf dpath2buf
prepresio s2ucmd
sectors @ dup >< daux !
$FFFF = if 1 daux ! then
bytsec @ $100 = if
    daux c@ $80 or daux c!
else
    daux c@ $7F and daux c!
then
$07 dcmd c! buf dbuf !
$c8 dtimlo c! writecmd
;

: rmatr
clearbuf dpath2buf
prepresio s2ucmd
$09 dcmd c!
$00 daux !
buf dbuf !
$c8 dtimlo c!
writecmd
;

: mountatr
clearbuf dpath2buf
prepresio s2ucmd
$BB daux c!
$05 dcmd c!
buf dbuf !
$c8 dtimlo c!
writecmd
;

: seccmd ( i cmd -- rc )
prepresio dcmd c!
dup 4 < if $80 else bytsec @ then dbyt !
daux ! secbuf @ dbuf !
;

: readsec ( i -- )
$52 seccmd devnm @ dunit c! readcmd
;

: writesec ( i -- )
$50 seccmd 3 dunit c! writecmd
;

code fullquery ( checks if sector buffer is empty )
[   lda secbuf   ]
[   sta ta       ]
[   lda secbuf+1 ]

```

```

[ sta ta+1 ]
[ ldy #0 ]
[ tya ]
end
code fql
[ ora (ta),y ]
[ iny ]
[ cpy bytsec ]
[ bne fql ]
[ jmp push0a ]
end

```

```
# user interface code
```

```

: clearfcount
  0 fcount !
;

```

```

: toggleautoinc
  1 autoinc ctoggle
  clearfcount
;

```

```

: qclearfcount ( n -- )
  fcount @ swap > if clearfcount then
;

```

```

: printfile
  cdigit @ 1 = if 9 qclearfcount then
  cdigit @ 2 = if 99 qclearfcount then
  cdigit @ 3 = if 999 qclearfcount then
  cdigit @ 4 = if 9999 qclearfcount then
  fcount @ >outbuf
  cdigit @ i ! {
    $30 ( ASCII 0 ) dpath dup c@ + 3 sidetgl @ + i @ + - c!
    i -- i @ 0 = if break then
  }
  $A1 ( source = outbuf+1 )
  dpath dup c@ + $A0 c@ 3 sidetgl @ + + - ( dest )
  $A0 c@ cmove
;

```

```

: nextfile
  cdigit @ if
  fcount ++
  printfile
then
;

```

```

: prevfile
  cdigit @ if
  fcount --
  fcount @ $FFFF = if 9999 fcount ! then
  printfile
then
;

```

```

: nextside
  1 side ctoggle

```



```

side @ if 65 else 66 then dpath dup c@ + 4 - c!
;

: toggleside
1 sidetgl ctoggle
sidetgl @ if nextside then
;

: printinfo
$00 sectors !
$80 bytsec !
18 secpertack !
stdensity disp colon mdquery 0 >
if stmedium disp
else
  getpercom
  percom 2+ @ >< tracks !
  percom 6 + @ 1 = if
    stdouble disp $100 bytsec !
  else
    stsingle disp
  then
then
space stsector disp
$73 emit colon
sectors @ u. space space
;

: printsource
5 2 at stsrc disp colon $44 emit
devnm c@ $30 + emit colon
space printinfo
;

: printdestination
crsoff 7 2 at stdest disp colon dpath disp " " disp
;

: printautoinc
crsoff 9 2 at stauto disp stincrement disp colon
autoinc @ if ston else stoff then disp
9 20 at stdigit disp colon
autoinc @ if cdigit @ else 0 then u.
9 30 at stside disp colon
sidetgl @ if ston else stoff then disp
;

: incdigit
cdigit ++
cdigit @ 5 = if
  0 autoinc c!
  0 cdigit !
else
  1 autoinc c!
then
;

: invertcursor ( i -- )
dup dpath + c@ $80 xor

```

```

swap dpath + c! printdestination
;

: inccursor
i @ dpath c@ < if i ++ then
;

: deccursor
i @ 1 > if i -- then
;

: newname
1 i !
{
i @ invertcursor
key
i @ invertcursor
dup 47 = if drop 92 then ( transpose / to \ )
dup $27 = if break then ( ESC )
dup $9B = if break then ( RETURN )
dup 42 = if inccursor then ( cursor right )
dup 43 = if deccursor then ( cursor left )
dup 157 = if dpath c@ 30 < if
i @ dpath + dup 1+ dpath c@ i @ - 1+ cmove> ( INSERT )
bl i @ dpath + c!
dpath c@ 1+ dpath c! ( dpath c++ )
then then
( dup $2F $59 within )
dup $2C > if dup $5D < if dup dpath i @ + c! inccursor then then ( ALPHANUMERI
dup 126 = if dpath c@ i @ - 0 > if ( DELETE )
i @ dpath + dup 1+ swap dpath c@ i @ - cmove
dpath c@ 1- dpath c! ( dpath c-- )
then then
125 = if 1 dpath c! 47 dpath 1+ c! then ( CLEAR )
}
drop
;

: beep
$FD emit
;

: clserr
19 lcls 19 lcls 19 lcls 19 lcls
;

: errorwaitkey
beep key drop clserr
;

: errorhead
20 1 at sterror disp space dstats c@ u. space
;

: prints2ustatus
21 2 at gets2ustatus "Sio2USB:" disp drvstat 15 + c@ dup u. $2F emit $24 emit dup .
22 2 at
dup $2E = if "Illegal characters in filename" disp then
dup $32 = if "USB device is not mounted" disp then

```

```

dup $34 = if "ATR filename not allowed" disp then
dup $35 = if "Directory not found" disp then
dup $37 = if "Filename already exists, overwrite?" disp then
dup $38 = if "Not enough free memory on storage" disp then
dup $39 = if "No free root directory entry found" disp then
drop
;

: printerr
errorhead
stwhile dispSPACE dispSPACE stsector dispSPACE
s2ucmdp if
prints2ustatus
then key drop
;

: secerr
errorhead
stillegal dispSPACE stsector dispSPACE stsize dispSPACE
key drop
;

: readerr
streadding printerr
;

: writeerr
stwriting printerr
;

: makeerr
errorhead
stwhile dispSPACE stcreating dispSPACE statr dispSPACE
prints2ustatus
beep key clserr
;

: rmerr
errorhead
stwhile dispSPACE stremoving dispSPACE statr dispSPACE
prints2ustatus
beep key clserr
;

: mounterr
errorhead
stwhile dispSPACE stmounting dispSPACE statr dispSPACE
prints2ustatus errorwaitkey
;

: resetcounter
stzeros0 stzeros1 strcpy
;

: checkhighspeed ( addr -- )
@ $E459 = 0if sthighspeed else stnormalspeed then disp
;

: setsecbufbase

```

```

appmhi @ $FF00 and $400 + bufbase !
$7000 bufbase @ - bufsize !
bufbase @ bufsize @ + bufend !
$7000 appmhi !
$7000 hsbases !
;

: readbuffer
hsdrive @ siov !
bufbase @ secbuf !
stzeros1 stzeros2 strcpy
sector @ i ! {
    i ++
   scopy 19 15 at stzeros1 disp debugflg @ if space secbuf @ .$ then # key drop
    i @ readsec errorcode c!
   bytsec @ secbuf +!
   sectors @ i @ = if break then # last sector
   bufend @ secbuf @ = if break then # buffer full?
    errorcode c@ $7F > if break then # error during read?
}
;

: writebuffer
hss2u @ siov !
bufbase @ secbuf !
stzeros2 stzeros1 strcpy
sector @ i ! {
    i ++
   scopy
    fullquery 0 = 0if
    19 15 at stzeros1 disp debugflg @ if space secbuf @ .$ then # key drop
    i @ writesecc errorcode c!
    then
   bytsec @ secbuf +!
   sectors @ i @ = if break then # last sector
   bufend @ secbuf @ = if break then # buffer full?
    errorcode c@ $7F > if break then # error during write?
}
i @ sector !
;

: copy
11 12 rcls
printsources
sectors @ not if
    secerr exit
then
19 2 at stmake disp space statr disp stdots disp
makeatr 1 > if
    makeerr $59 = 0if
    exit
else
    clserr
    19 2 at "removing" disp space statr disp stdots disp
    rmatr 1 > if rmerr exit then
    makeatr 1 > if makeerr exit then
then
then
19 2 at stmount disp space statr disp stdots disp

```

```

mountatr 1 > if mounterr exit then
19 8 at stsector disp space stzeros1 disp
15 2 at "Drive " disp devnm c@ $30 + emit space hsdribe checkhighspeed
debugflg @ if space hsdribe @ .$ then
16 2 at "SIO2USB " disp hss2u checkhighspeed
debugflg @ if space hss2u @ .$ then
debugflg @ if
    12 2 at "BufferSize:" disp space bufsize @ .$
    13 2 at "Bufend:" disp space bufend @ .$
    14 2 at "APPMHI:" disp space appmhi @ .$
then
bufbase @ secbuf !
resetcounter
1 errorcode !
0 sector !
{
    19 2 at stread disp
    readbuffer errorcode c@ 1 > if readerr then
    errorcode c@ $80 < if
        19 2 at stwrite disp
        writebuffer errorcode c@ 1 > if writeerr then
        then
        sectors @ sector @ = if break then
        errorcode c@ $7F > if break then
    }
$E459 siov ! ( reset to original SIOV )
clserr stcopy disp space
errorcode c@ $80 < if stsuccessfully else stunsucessful then disp space
stcompleted disp space cr cr stnextcpy disp space
autoinc @ if
    sidetgl @ if
        side @ if nextside else nextside nextfile then
    else
        nextfile
    then
        printdestination
then
;

: status
11 12 rcls
getstatus drop 11 2 at printstatus
getpercom drop 14 2 at printpercom
;

: type ( caddr u -- )
{
    swap dup c@ emit 1+ swap 1-
    dup 0 = if drop drop break then
}
;

: printdirentry
cnt @ 11 +
dup 20 > if 10 - 22 else 2 then at
$10 * bufbase @ +
dup @ 0 = if drop exit then
dup @ $80 and if drop exit then
cnt ++

```

```

dup @ $20 and if $2A emit else space then space
( 1+ dup @ .$ space ) ( Length )
( 2+ dup @ .$ space ) ( Start Sector )
5 + dup 8 type ( Filename )
    $2e emit ( . )
8 + 3 type cr ( Extender )
;

: showdirectory
11 12 rcls
0 cnt !
bufbase @ secbuf !
360 sector !
{
  1 sector +!
  sector @ readsec drop
  0 i !
  {
    cnt @ 22 = if 22 2 at " press key" disp key drop 0 cnt ! 11 12 rcls then
    i @ printdirentry
    i ++
    i @ $40 = if break then
  }
  sector @ 368 = if break then
}
22 2 at " press key" disp key drop
;

: ver
"1.2p2 SIO2USB" disp
;

: banner
cls
$90 $2C6 c! crsoff
statr disp stcopy disp stcenter disp
" (c) 2014 C. Strotmann" disp cr
"Version " disp ver
printsource 4 line
printdestination 6 line
printautoinc 8 line
10 line
1 24 at sthelpmsg disp
;

: help
11 12 rcls
12 2 at "1-9 : select source and query disk" disp
13 2 at " / : edit ATR filename" disp
14 2 at " A : toggle autoincrement" disp
15 2 at " + : increment digits" disp
16 2 at " B : enable/disable diskside A/B" disp
17 2 at " N/P : inc-/decrement number" disp
18 2 at " X : toggle diskside A/B" disp
19 2 at " S : print status" disp
20 2 at " D : print DOS 2.x directory" disp
21 2 at " Q : quit program" disp
22 0 at " [SPACE] copy source disk to ATR" disp
;

```

```

: acc
  setsecbufbase
  l devnm c! getdrvsio

temp2dpath banner help
{
  crsoff key crsoff
  dup $30 > if
    dup $3A < if
      dup $30 - devnm c! printsource getdrvsio
    then
  then
  dup $4E = if nextfile printdestination then
  dup $50 = if prevfile printdestination then
  dup $48 = if help then
  dup $20 = if copy then
  dup $3F = if help then
  dup $54 = if temp2dpath then
  ( dup $44 = if newpath 0 fcount ! 7 clearline printdestination then )
  dup $2F = if newname 0 fcount ! printdestination then
  dup $41 = if toggleautoinc printautoinc then
  dup $42 = if toggleside printautoinc printdestination then
  dup $2B = if incdigit printautoinc then
  dup $53 = if status then
  dup $44 = if showdirectory help then
  dup $58 = if sidetgl @ if nextside printdestination then then
  dup $51 = if exit then
  dup $7D ( CLEAR ) = if banner then
  dup $3F = if debugflg ctoggle 1 38 at debugflg @ if $44 else $20 then emit then
  drop
}
  cls crson
;

: main
  init
  acc
  bye
;

```